Module 1

- **Bisect^{n} Method**

$$x_n = \frac{x_{n-1} + x_{n-2}}{2}$$

  - 2 pt. method
  - Rate of convergence is 1.
  - n= 2, 3, 4,...

- **Regula-Falsi (False Posit^{n}) Method**

$$x_1 = a - \left[f(a) \left(\frac{b-a}{f(b) - f(a)} \right) \right]$$

  - 2 pt. method
  - Rate of convergence is 1.
  - **1st approx. lying between (a, x□)**

$$x_{n+1} = a - \left[f(a) \left(\frac{x_n - a}{f(x_n) - f(a)} \right) \right]$$

    n= 1,2,3,...

  - **1st approx. lying between (x□, b)**

$$x_{n+1} = b - \left[f(b) \left(\frac{b - x_n}{f(b) - f(x_n)} \right) \right]$$

    n=1,2,3,...

- **Secant Method**

$$x_{n+1} = x_n - \left[f(x_n) \left(\frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} \right) \right]$$

  - 2 pt. method
  - Rate of convergence is 1.618 =(1+√5)/4
  - n= 1,2,3,...

- Newt^{n}-Raphs^{n} Method

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

• 1 pt. method
• Rate of convergence is 2.
• n= 1,2,3,...

- Fixed-pt. method

$$x_{n+1} = \varphi(x_n)$$

• 1 pt. method
• Rate of convergence is 1.

  - Condit^{n}

$$|\varphi'(x)| < 1$$

- Indirect Methods

  - Gauss-Seidel Method

$$x^{(k+1)} = \frac{1}{a_1}(d_1 - b_1 y^{(k)} - c_1 z^{(k)})$$
$$y^{(k+1)} = \frac{1}{b_2}(d_2 - a_2 x^{(k+1)} - c_2 z^{(k)})$$
$$z^{(k+1)} = \frac{1}{c_3}(d_3 - a_3 x^{(k+1)} - b_3 y^{(k+1)})$$

k= 0,1,2,3,...

  - Condit^{n}

$$|a_1| > |b_1| + |c_1|$$
$$|b_2| > |a_2| + |c_2|$$
$$|c_3| > |a_3| + |b_3|$$

- **N.B.**
  - Also called **Liebmann method** or **method of successive displacement**.
  - Gauss-Seidel method is a particular case of SOR method, when ω=1.

- **SOR Method**

$$x^{(k+1)} = (1-\omega)x^{(k)} + \frac{\omega}{a_1}(d_1 - b_1 y^{(k)} - c_1 z^{(k)})$$

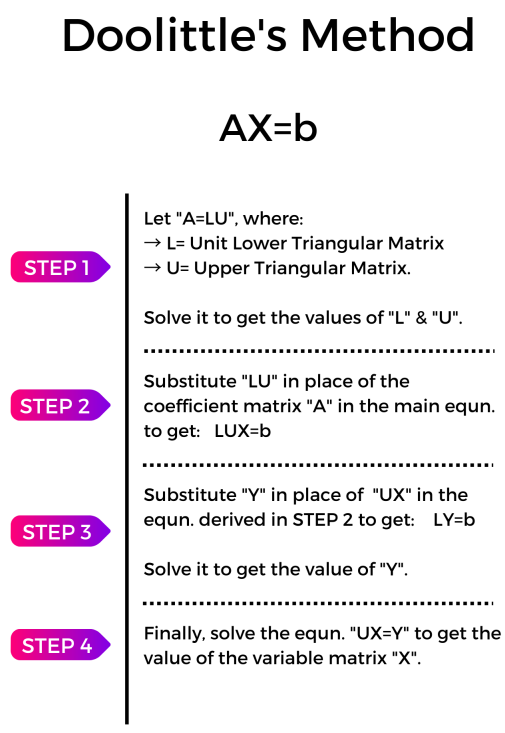
$$y^{(k+1)} = (1-\omega)y^{(k)} + \frac{\omega}{b_2}(d_2 - a_2 x^{(k+1)} - c_2 z^{(k)})$$

$$z^{(k+1)} = (1-\omega)z^{(k)} + \frac{\omega}{c_3}(d_3 - a_3 x^{(k+1)} - b_3 y^{(k+1)})$$

k= 0,1,2,3,...

- **Condit^n**
  - Coefficient matrix must be **symmetric** & **+ve definite**.
  - ω ∈ (0, 2)

- **Direct Methods**
  - **Doolittle's Method**


Doolittle's Method
AX=b
STEP 1
Let "A=LU", where:
→ L= Unit Lower Triangular Matrix
→ U= Upper Triangular Matrix.
Solve it to get the values of "L" & "U".
STEP 2
Substitute "LU" in place of the coefficient matrix "A" in the main equn. to get: LUX=b
STEP 3
Substitute "Y" in place of "UX" in the equn. derived in STEP 2 to get: LY=b
Solve it to get the value of "Y".
STEP 4
Finally, solve the equn. "UX=Y" to get the value of the variable matrix "X".

  - **Crout's Method**

Crout's Method

AX=b

STEP 1
Let "A=LU", where:
→ L= Lower Triangular Matrix
→ U= Unit Upper Triangular Matrix.

Solve it to get the values of "L" & "U".

STEP 2
Substitute "LU" in place of the coefficient matrix "A" in the main equn. to get: LUX=b

STEP 3
Substitute "Y" in place of "UX" in the equn. derived in STEP 2 to get: LY=b

Solve it to get the value of "Y".

STEP 4
Finally, solve the equn. "UX=Y" to get the value of the variable matrix "X".

- Cholesky Method

Cholesky Method

AX=b

Here the Coefficient matrix must be symmetric & +ve definite.

STEP 1
Let "$A=LL^T$", where:
→ L= Lower Triangular Matrix
→ L^T= Transpose of "L" Matrix.

Solve it to get the values of "L" & "L^T".

STEP 2
Substitute "LL^T" in place of the coefficient matrix "A" in the main equn. to get: $LL^TX=b$

STEP 3
Substitute "Y" in place of "L^TX" in the equn. derived in STEP 2 to get: LY=b

Solve it to get the value of "Y".

STEP 4
Finally, solve the equn. "$L^TX=Y$" to get the value of the variable matrix "X".

- Interpolation
  - Lagrange's IP

$$P_n(x) = \sum_{i=0}^{n} l_i(x) \cdot f(x_i)$$

$$l_i(x) = \frac{\prod_{k=0}^{n} (x - x_k)}{\prod_{k=0}^{n} (x_i - x_k)}, \quad (k \neq i)$$

- **N.B.**

$$l_i(x_i) = 1$$

$$l_i(x_j) = \begin{cases} 0 & ,i \neq j \\ 1 & ,i = j \end{cases}$$

- **NDDIP**

$$f(x_0) + \sum_{i=1}^{n} \left[\left[\prod_{k=1}^{i} (x - x_{k-1}) \right] f[x_0, x_1, x_2, \ldots, x_i] \right]$$

- **NFDIP**

$$f(x_0) + \sum_{i=1}^{n} \left[\left[\prod_{k=1}^{i} (x - x_{k-1}) \right] \frac{\Delta^i f(x_0)}{i! \, . \, h^i} \right]$$

  - **N.B.**

    When the forward differences in the forward difference table are rapidly increasing, in this case, the interpolating polynomial doesn't fit to the funct^n at all.

- **NBDIP**

$$f(x_n) + \sum_{i=1}^{n} \left[\left[\prod_{k=n}^{n+(1-i)} (x - x_k) \right] \frac{\nabla^i f(x_n)}{i! \, . \, h^i} \right]$$

- **N.B.**

  The interpolating polynomial of a funct^n is unique.